



SDD

System Design Document

Sapori di Unisa

Riferimento	nc01_saporidiunisa-sdd.docx
Versione	1.2
Data	01/02/2023
Destinatario	Prof. Carmine Gravino
Presentato da	Antonio Facchiano, Gianmarco Riviello, Salvatore Ruocco, Simone Vittoria
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
23/11/2023	0.1	Stesura dell'introduzione	Gianmarco Riviello
28/11/2023	0.2	Suddivisione in sottosistemi	Salvatore Ruocco
29/11/2023	0.3	Mapping Hardware/Software	Salvatore Ruocco
29/11/2023	0.4	Design Goals	Antonio Facchiano
29/11/2023	0.5	Gestione dati persistenti	Gianmarco Riviello
30/11/2023	0.6	Architettura del sistema corrente	Gianmarco Riviello
30/11/2023	0.7	Servizi dei sottosistemi	Salvatore Ruocco
30/11/2023	0.7	Glossario	Gianmarco Riviello
03/12/2023	0.7	Revisione	Tutto il team
04/12/2023	0.8	Condizioni limite	Simone Vittoria
04/12/2023	0.9	Trade-off	Simone Vittoria
14/11/2023	1.0	Revisione e delibera del documento.	Facchiano Antonio; Riviello Gianmarco; Ruocco Salvatore; Vittoria Simone.



30/01/2024	1.1	Applicazione correzioni	Salvatore Ruocco, Gianmarco Riviello
01/02/2024	1.2	Revisione e delibera del documento.	Facchiano Antonio; Riviello Gianmarco; Ruocco Salvatore; Vittoria Simone.



Sommario

Sommario	4
1 Introduzione	5
1.1 Scopo del Sistema.....	5
1.2 Obiettivi di design (Design Goals)	5
1.3 Definizioni, acronimi e abbreviazioni	8
1.4 Riferimenti.....	8
1.5 Organizzazione del documento.....	8
2 Architettura del Sistema Corrente	9
3 Architettura del Sistema Proposto.....	9
3.1 Panoramica.....	9
3.2 Decomposizione in sottosistemi	10
3.3 Mapping hardware/software	12
3.4 Gestione dei dati Persistenti	13
3.5 Controllo degli accessi e sicurezza	20
3.6 Controllo globale del software	20
3.7 Condizioni limite	21
4 Servizi dei sottosistemi.....	24
5 Glossario	27



1

Introduzione

1.1 Scopo del Sistema

Sapori di Unisa si propone di semplificare e ottimizzare attività quotidiane di un supermercato e di conseguenza ridurre il più possibile gli errori umani.

Il sistema è gestito da un Admin, il cui compito è assicurare la sicurezza e garantire l'esclusività delle operazioni al solo personale qualificato.

Il sistema è invece utilizzato dai dipendenti del supermercato: magazzinieri, cassieri e addetti alle finanze. Queste figure svolgono le loro mansioni attraverso il sistema.

1.2 Obiettivi di design (Design Goals)

In questa sezione verranno presentati i design goals, i quali rappresentano le qualità su cui il sistema dovrà focalizzarsi.

Seguendo le linee guida, i design goals si suddividono nelle seguenti categorie.

- **Performance:** includono i requisiti di spazio e velocità imposti sul sistema.
- **Dependability:** determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, falle di sicurezza) e le loro conseguenze.
- **Maintenance:** determina quanto sforzo è necessario per modificare il sistema dopo il suo rilascio.
- **End User:** includono qualità che sono desiderabili dal punto di vista dell'utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Ciascun design goal è descritto dai seguenti parametri.

- **Rank**, che ne specifica un valore di priorità compreso tra 1 e il numero di design goals individuati.
- **ID**, un identificatore univoco e un nome esplicativo.
- **Descrizione**, una descrizione del design goal.
- **Categoria**, ovvero la categoria di appartenenza del design goal.
- **RNF**, che specifica il requisito che lo ha generato.

Design Goals

Rank	ID	Descrizione	Categoria	RNF
5	DG_1 Facilità d'uso	Il sistema deve risultare comprensibile attraverso l'uso delle "8 regole d'oro di Shneiderman" per il design delle interfacce grafiche.	End User	RNF_U_3 RNF_U_1



6	DG_2 Feedback immediato	Il sistema deve restituire un feedback chiaro ad ogni interazione dell'utente, in modo che questi possa comprendere facilmente il funzionamento del sistema.	End User	RNF_U_2
7	DG_3 Affidabilità delle operazioni	Il sistema deve garantire che tutte le operazioni avvengano con successo.	Dependability	RNF_A_1
1	DG_4 Sicurezza degli utenti	Il sistema deve garantire l'esecuzione di operazioni solo da utenti con i permessi necessari, al fine di mantenere integro il sistema.	Dependability	RNF_A_2
2	DG_5 Gestione dei fallimenti	Il sistema deve essere resiliente ad errori e fallimenti che possono verificarsi durante l'utilizzo da parte dell'utente, notificando quest'ultimo tramite dei messaggi.	Dependability	RNF_A_3
3	DG_6 Tempi di risposta	Il sistema deve garantire tempi di risposta di non oltre 10 secondi, anche in condizione di più utenti connessi contemporaneamente.	Performance	RNF_P_1 RNF_P_3
4	DG_7 Disponibilità delle informazioni	Il sistema deve garantire la fruizione di tutte le informazioni negli orari di lavoro prestabiliti.	Performance	RNF_P_2 RNF_P_5
8	DG_8 Responsiveness	Il sistema deve mostrare un'interfaccia in grado di potersi adattare ad ogni tipo di schermo.	Performance	RNF_P_4

9	DG_9 Facilità di estensione	Il sistema deve essere sviluppato in modo tale da poter accogliere eventuali tecnologie future, seguendo gli standard IEEE e ISO, in modo da poter estendere le sue funzionalità.	Maintenance	RNF_S_1 RNF_S_2 RNF_IM_3
10	DG_10 Utilizzo hardware e software	Il sistema deve essere sviluppato come una piattaforma web e deve funzionare su hardware già disponibile al momento della progettazione dello stesso.	Maintenance	RNF_IM_1 RNF_IM_2

Trade-off

Trade-off	Descrizione
Facilità d'uso vs complessità	L'introduzione di funzionalità aggiuntive con una complessità ed un livello di dettaglio maggiori, accrescerebbe la completezza del sistema. Tuttavia, ciò renderebbe il sistema più difficile da utilizzare e da comprendere per gli utenti meno esperti.
Gestione dei fallimenti vs tempi di risposta	Avere procedure complesse di gestione dei fallimenti potrebbe rallentare la risoluzione di problemi e influire sui tempi di risposta.
Riservatezza vs condivisione dei dati	Effettuare una netta distinzione tra le operazioni permesse a ciascun utente, può rendere più difficile la comunicazione e la condivisione dei dati tra gli utenti del sistema.



1.3 Definizioni, acronimi e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **Sottosistema:** un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.
- **Dati Persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.
- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **SDD:** System Design Document.
- **RAD:** Requirements Analysis Document.
- **DG:** Design Goal
- **UC_BC:** Use Case Boundary Condition

1.4 Riferimenti

- Libro di testo utilizzato durante il corso: [Object-Oriented Software Engineering Using UML, Patterns, and Java](#)
- [Statement Of Work](#)
- [Requirements Analysis Document](#)

1.5 Organizzazione del documento

- **Introduzione:** Viene descritto in generale lo scopo del sistema e gli obiettivi di design che il sistema propone di raggiungere.
- **Architettura software corrente:** Viene descritto lo stato attuale dell'architettura del software già presente.
- **Architettura software proposta:** Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.
- **Glossario:** Contiene la lista dei termini usati nel documento con annessa spiegazione.



2

Architettura del Sistema Corrente

Sul mercato sono presenti alcune piattaforme simili a Saponi di Unisa che, fra i loro servizi offerti, presentano la gestione del magazzino, con particolare attenzione alle scorte e alle finanze. I prodotti in questione sono: [EuroAzienda Supermercato](#) e [Software FoodManager](#).

Da questi 2 prodotti si può evincere quella che è in parte l'idea da realizzare ma, a differenza loro che puntano molto sulla fidelizzazione del cliente, Saponi di Unisa si focalizza invece sulla tracciabilità dei dati all'interno del supermercato, sul minimizzare gli errori e le incongruenze fra magazzino e vendite, sul massimizzare quelli che sono i profitti tenendo sott'occhio le scadenze dei prodotti.

Da sottolineare, inoltre, che Saponi di Unisa non è finalizzato alla gestione dei clienti e che quindi si differenzia anche in questo con i propri competitors.

3 Architettura del Sistema Proposto

3.1 Panoramica

L'architettura del sistema è di tipo **three-tier**. Uno dei motivi che hanno influenzato tale scelta è sicuramente il background degli sviluppatori, i quali hanno già avuto a che fare con tale architettura. Inoltre, essa è molto indicata per lo sviluppo di applicazioni web, poiché si ha una netta separazione tra interfacce utente, logica applicativa e persistenza dei dati. Queste caratteristiche garantiscono una serie di qualità essenziali per lo sviluppo software odierno, su tutte la manutenzione e il riuso.

Per lo sviluppo del sistema verranno usate le seguenti tecnologie:

Back-end	Front-end
<ul style="list-style-type: none">• Java 21• MySQL 5.6• Apache Tomcat 10.1	<ul style="list-style-type: none">• HTML5• JSP 3.1• CSS3• JavaScript

Per la gestione delle librerie verrà usato **Maven**.

3.2

Decomposizione in sottosistemi

Abbiamo deciso di dividere il sistema nei seguenti sottosistemi.

- **Gestione Finanze:** offre servizi che riguardano gli aspetti finanziari del sistema: impostare sconti, visualizzare il bilancio, controllare l'andamento delle vendite dei prodotti.
- **Gestione Magazzino:** offre servizi che riguardano la gestione delle forniture e del magazzino: registrare nuove forniture, eliminare lotti, visualizzare prodotti del magazzino.
- **Gestione Scaffali:** offre servizi per i prodotti esposti sugli scaffali: aggiunge prodotti agli scaffali, elimina i prodotti scaduti dagli scaffali, visualizza i prodotti presenti sugli scaffali.
- **Gestione Vendite:** offre servizi che riguardano la vendita dei prodotti alla clientela: conferma le spese dei clienti, visualizza lo storico delle vendite.
- **Autenticazione:** offre servizi che riguardano l'autenticazione al sistema: login e logout degli utenti, modifica dei pin di accesso.
- **StorageConnection:** intermezzo tra lo Storage e gli altri sottosistemi.
- **Storage:** responsabile del database e della gestione dei dati persistenti.

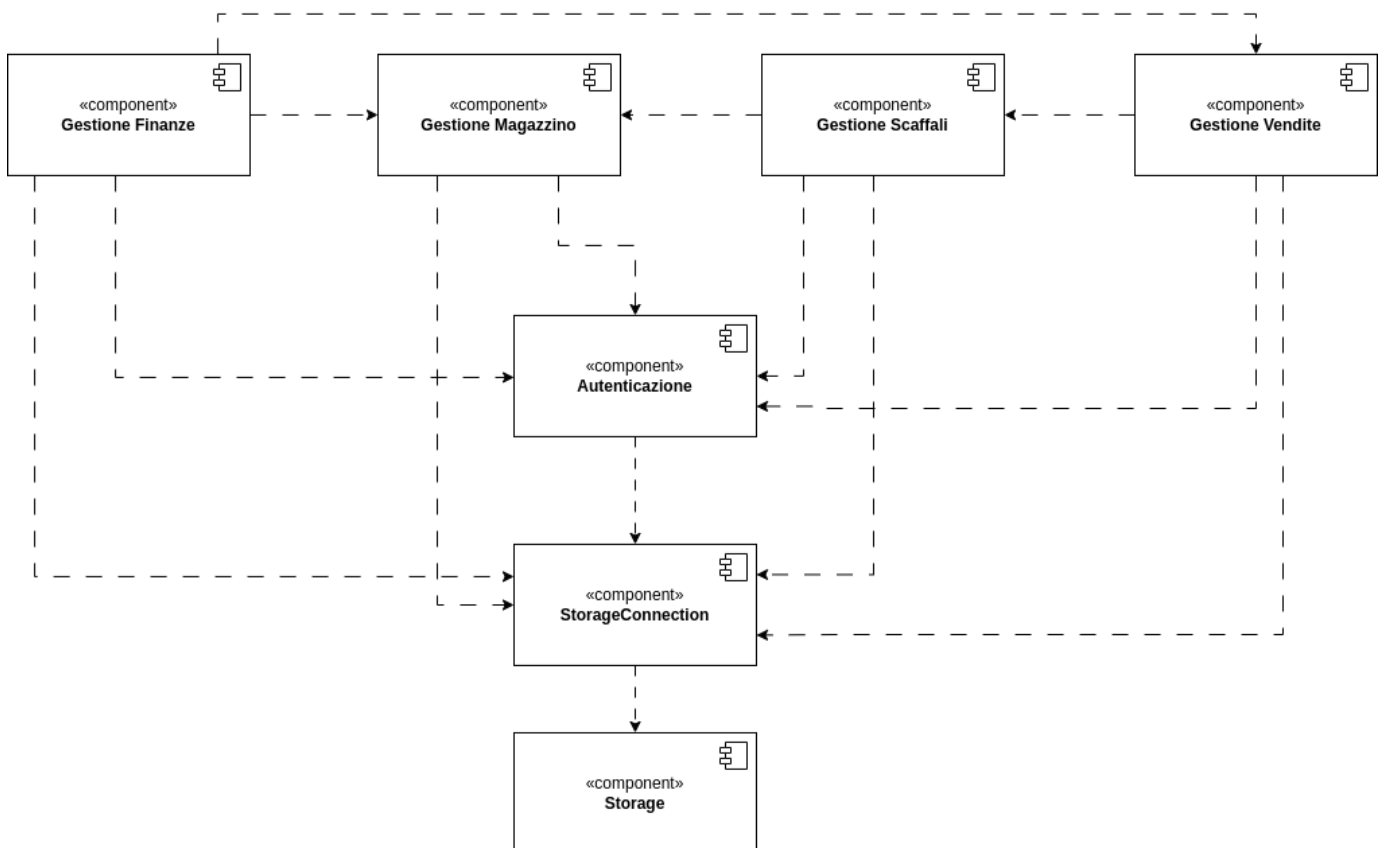
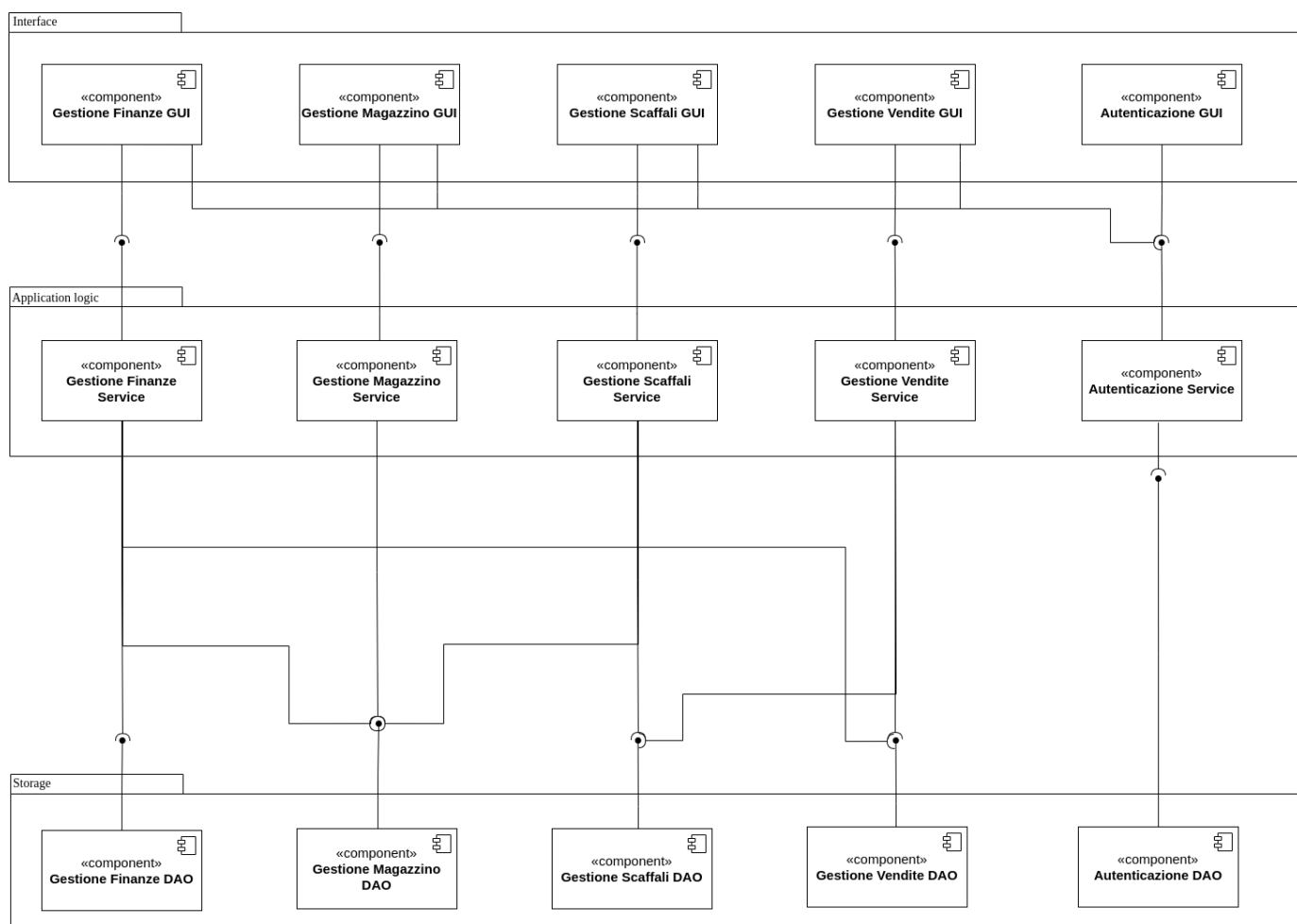


Diagramma architettura – UML Component Diagram



Interface

Tutte le componenti di questo livello utilizzano Autenticazione Service per poter usufruire del login e del logout.

Application Logic

Gestione Finanze Service può accedere ai dati di:

- Gestione Magazzino DAO per ottenere informazioni riguardo le forniture scaricate.
- Gestione Vendite DAO per ottenere informazioni riguardo le vendite effettuate.

Gestione Scaffali Service può accedere ai dati di:

- Gestione Magazzino DAO per ottenere informazioni riguardo ai prodotti presenti in magazzino.

Gestione Vendite Service può accedere ai dati di:

- Gestione Scaffali DAO per ottenere informazioni riguardi ai prodotti presenti sugli scaffali.

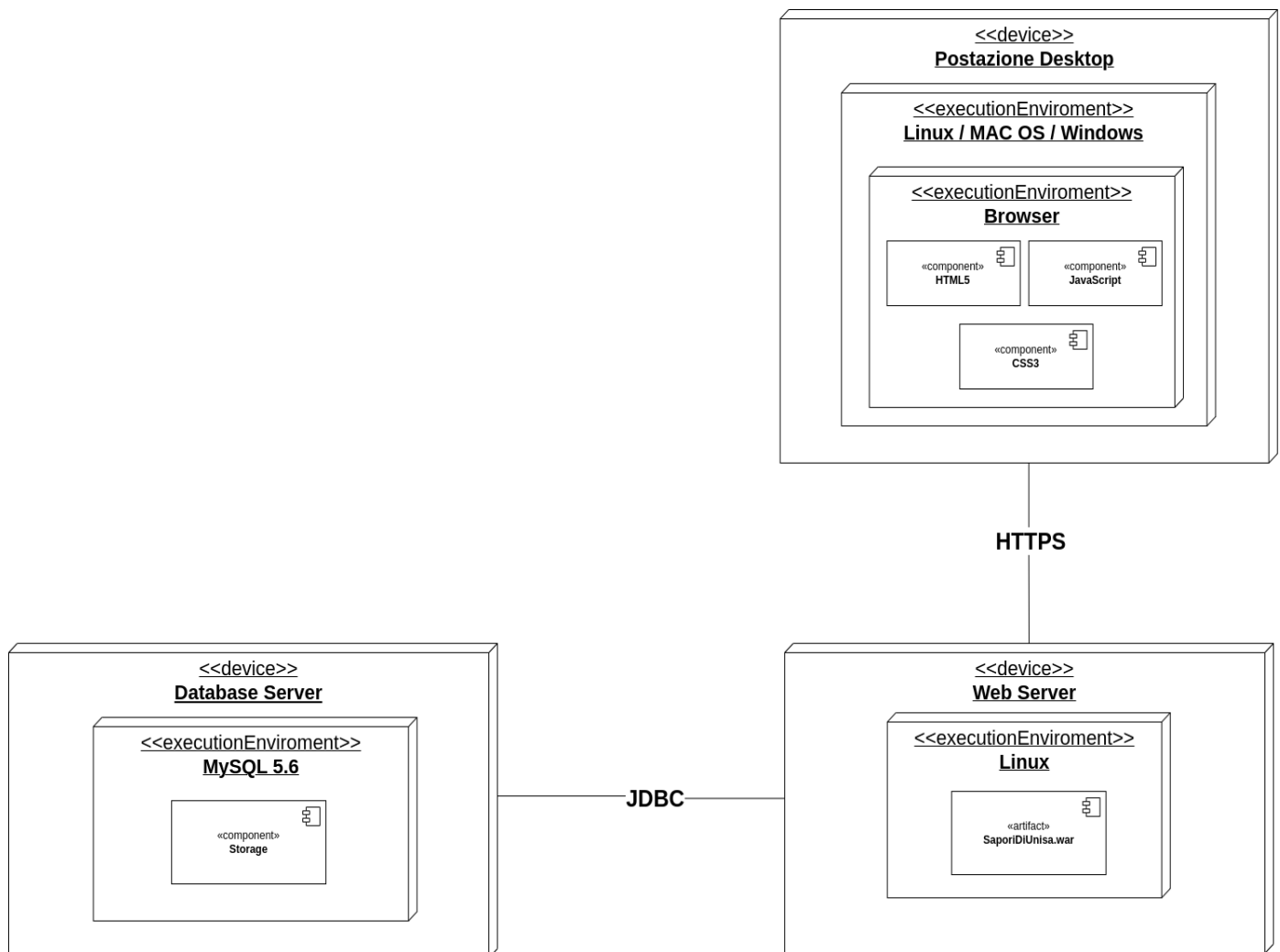


3.3 Mapping hardware/software

L'applicazione web che si intende realizzare sarà costituita da un server, risiedente su un solo nodo, che risponderà alle richieste di un client. Quest'ultimo, mediante l'interfaccia fornita dal browser, potrà effettuare operazioni col sistema mediante una richiesta HTTPS, alla quale il server procederà a rispondere con la funzionalità richiesta.

Il seguente Deployment Diagram descrive il mapping tra hardware e software.

UML Deployment Diagram



3.4 Gestione dei dati Persistenti

Per la gestione dei dati persistenti del sistema, si è optato di utilizzare un database di tipo relazionale in modo da gestire agevolmente l'accesso ai dati e garantire la consistenza tramite l'utilizzo di un DBMS.

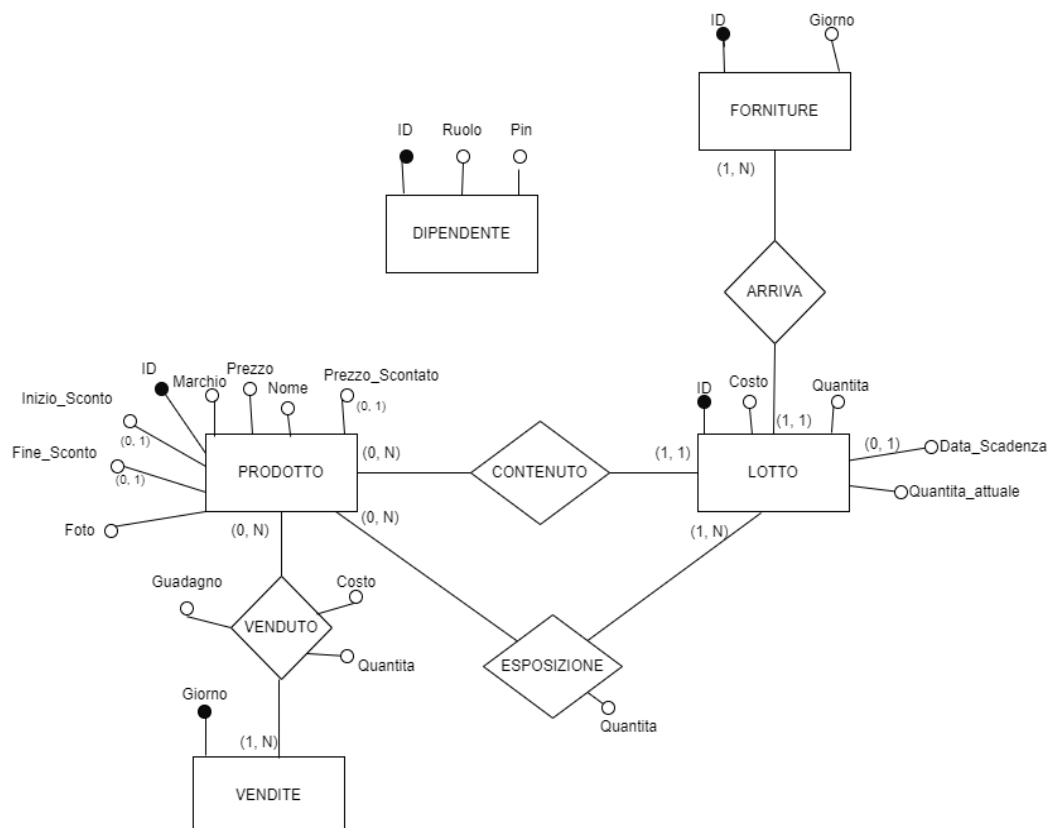
Questa scelta è stata presa tenendo conto degli obiettivi prefissati, per l'alta affidabilità garantita dal DBMS e le competenze maturate nel corso degli anni da tutti i membri del Team.

Nello specifico si è deciso di usare un prodotto messo a disposizione dalla **Oracle Corporation**. Il database è quindi: **MySQL**.

Le caratteristiche messe a disposizione da questo prodotto sono:

- **Imposizioni di vincoli di integrità sui dati:** il DBMS permette di definire diversi vincoli per mantenere l'integrità dei dati e controlla esso stesso che tali vincoli siano rispettati.
- **Privatezza dei dati:** il DBMS permette un accesso protetto e viste diverse sui dati in base ai vari tipi di utenti.
- **Affidabilità dei dati:** il DBMS mette a disposizione alcuni metodi per salvare copie di backup e per ripristinare lo stato del database in caso di guasti.
- **Atomicità delle operazioni:** il DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati. L'atomicità delle transazioni garantisce di mantenere uno stato della base di dati consistente.

Modello ER del Database:





Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof. Carmine Gravino



Dizionario dei dati

In questa sezione del documento si mostrano gli attributi dello schema logico derivante dall'ER.

Nome Entità	Dipendente		
Descrizione	Contiene pin di accesso e nome del ruolo.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
ID	INT (32)	PRIMARY KEY	NOT NULL; AUTO_INCREMENT
ruolo	ENUM ('Cassiere', 'Admin', 'Magazziniere', 'Finanze')		NOT NULL
Pin	CHAR(4)		NOT NULL;

Nome Entità	Fornitura		
Descrizione	Contiene il giorno dell'avvenuta fornitura e il suo Identificativo.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
ID	INT (32)	PRIMARY KEY	NOT NULL; AUTO_INCREMENT
giorno	Date (AAAA-MM-GG)		NOT NULL



Nome Entità	Prodotto		
Descrizione	Descrive un prodotto, contiene le informazioni che lo riguardano e che servono all'interno del software.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
ID	INT (32)	PRIMARY KEY	NOT NULL; AUTO_INCREMENT
nome	Varchar(255)		NOT NULL
marchio	Varchar (255)		NOT NULL
prezzo	Decimal (5, 2)		NOT NULL; CHECK prezzo > 0.00
prezzo_scontato	Decimal (5, 2)		CHECK prezzo_scontato > 0.00 AND prezzo_scontato < prezzo
inizio_sconto	Date (AAAA-MM-GG)		CHECK inizio_sconto AFTER currentdate.
fine_sconto	Date (AAAA-MM-GG)		
foto	MediumBlob		NOT NULL



Nome Entità	Lotto		
Descrizione	Contiene il lotto scaricato dalla fornitura con relativi dati e rappresenta anche la quantità totale e attuale in magazzino.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
ID	INT (32)	PRIMARY KEY	NOT NULL; AUTO_INCREMENT
costo	Decimal (12, 2)		NOT NULL; CHECK costo > 0.0
data_scadenza	Date (AAAA-MM-GG)		
quantita	INT (6)		NOT NULL; CHECK quantita > 0
quantita_attuale	INT (6)		NOT NULL; CHECK quantita_attuale >= 0
<u>fornitura</u>	INT (32)	FOREIGN KEY fornitura REFERENCES Fornitura (id)	NOT NULL; ON DELETE CASCADE
<u>prodotto</u>	INT (32)	FOREIGN KEY prodotto REFERENCES Prodotto (id)	NOT NULL; ON DELETE CASCADE



Nome Entità	Vendita		
Descrizione	Contiene il giorno di lavoro		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
giorno	Date (AAAA-MM-GG)	PRIMARY KEY	NOT NULL;

Nome Entità	Esposizione		
Descrizione	Associazione fra Lotto e Prodotto che tiene conto dei prodotti di un determinato lotto messi in esposizione e delle loro quantità.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
<u>prodotto</u>	INT (32)	PRIMARY KEY, FOREIGN KEY prodotto REFERENCES Prodotto (id)	NOT NULL; ON DELETE CASCADE
<u>lotto</u>	INT (32)	PRIMARY KEY, FOREIGN KEY lotto REFERENCES Lotto (id)	NOT NULL; ON DELETE CASCADE
quantita	INT		NOT NULL; CHECK quantita >= 0



Nome Entità	Venduto		
Descrizione	Associazione fra Prodotto e Vendite che tiene conto dei prodotti venduti in un determinato giorno aggiornando le loro quantità, costi e guadagni durante la giornata.		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
costo	Decimal (5, 2)		NOT NULL; CHECK costo > 0.00
quantità	INT (6)		NOT NULL; CHECK quantità > 0
guadagno	Decimal (5, 2)		NOT NULL;
<u>prodotto</u>	INT (32)	FOREIGN KEY prodotto REFERENCES Prodotto (id)	NOT NULL; ON DELETE CASCADE
giorno	Date (AAAA-MM-GG)	FOREIGN KEY Vendite REFERENCES Vendite (Giorno)	NOT NULL;



3.5

Controllo degli accessi e sicurezza

Attori	Magazziniere	Addetto alle finanze	Cassiere	Admin
Oggetti				
Gestione Finanze		<ul style="list-style-type: none">- Visualizza Bilancio- Imposta Sconti- Visualizza prodotti- Visualizza andamento prodotto		
Gestione Magazzino	<ul style="list-style-type: none">- Registra fornitura- Visualizza prodotti magazzino- Visualizza forniture			<ul style="list-style-type: none">- Eliminazione lotto- Visualizza forniture- Modifica fornitura
Gestione Scaffale	<ul style="list-style-type: none">- Aggiungi prodotti scaffale- Visualizza prodotti scaffale- Elimina scaduto scaffale		<ul style="list-style-type: none">- Visualizza prodotti scaffale	
Gestione Vendite			<ul style="list-style-type: none">- Vendita prodotti- Visualizza storico vendite	
Autenticazione	<ul style="list-style-type: none">- Login- Logout	<ul style="list-style-type: none">- Login- Logout	<ul style="list-style-type: none">- Login- Logout	<ul style="list-style-type: none">- Login- Logout- Modifica Pin

3.6 Controllo globale del software

Il flusso del sistema fornisce diverse funzionalità che richiedono una continua interazione da parte dell'utente, ragione per cui, il controllo del flusso globale è di tipo **event-driven** ovvero guidato dagli eventi. Quando si verifica un evento, vengono selezionati i sottosistemi che si occupano della logica applicativa e della gestione dei dati persistenti.

3.7

Condizioni limite

Identificativo <i>UC_BC_1</i>		<i>Data</i>	04/12/2023
		<i>Vers.</i>	1.0
		<i>Autore</i>	Simone Vittoria
Descrizione	Il seguente UC descrive l’avvio del sistema		
Attore Principale	Admin		
Attori secondari	NA		
Entry Condition	Il server sul quale risiede il sistema è in funzione AND L’admin accede al server utilizzando le credenziali di accesso		
Exit condition On success	Il sistema è in esecuzione sul server ed è pronto a gestire le richieste degli utenti		
Exit condition On failure	Il sistema non viene eseguito e dunque non può operare		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Admin	Esegue le operazioni necessarie all’avvio del sistema sul server	
2	Server	Verifica la disponibilità di aggiornamenti per il sistema. Verifica l'integrità dei dati persistenti. Infine, avvia il sistema verificando la disponibilità e il corretto funzionamento dei suoi servizi.	
Scenario/Flusso di eventi Alternativo: Sono presenti aggiornamenti del sistema			
2.a1	Server	Non effettua l’avvio del sistema. Notifica l’amministratore con un messaggio, riportando gli aggiornamenti disponibili.	
2.a2	Admin	Seleziona l’opzione per aggiornare il sistema.	
2.a3	Server	Esegue gli aggiornamenti disponibili e avvia il sistema automaticamente.	
Scenario/Flusso di eventi Alternativo: I dati persistenti sono danneggiati o mancanti			
2.b1	Server	Non effettua l’avvio del sistema. Notifica l'amministratore con un messaggio di errore riportando l’errore che si è verificato e la possibilità di eseguire il ripristino da backup.	
2.b2	Admin	Esegue il ripristino dati da backup. Prova a riavviare il sistema ripartendo dal passo 1.	
Scenario/Flusso di eventi di Errore: Fallimento di uno o più servizi del sistema			
2.c1	Server	Dopo aver avviato il sistema, verifica il corretto funzionamento dei suoi servizi. Se uno dei servizi fallisce, il server interrompe l’esecuzione del sistema e avvisa l'amministratore con un file di log.	
2.c2	Admin	Esamina i log per identificare la causa del fallimento. Corretti eventuali problemi, prova a riavviare il sistema ripartendo dal passo 1.	



Identificativo <i>UC_BC_2</i>	<i>Spegnimento del sistema</i>	<i>Data</i>	04/12/2023
		<i>Vers.</i>	1.0
		<i>Autore</i>	Simone Vittoria
Descrizione	Il seguente BC descrive lo spegnimento del sistema		
Attore Principale	Admin		
Attori secondari	NA		
Entry Condition	Il server è in funzione AND Il sistema è in esecuzione sul server AND Non sono in esecuzione processi critici durante i quali il sistema non può essere arrestato AND L'admin accede al server utilizzando le credenziali di accesso		
Exit condition On success	Il sistema in esecuzione sul server viene spento correttamente		
Exit condition On failure	Il sistema non viene spento		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Admin	Esegue le operazioni necessarie allo spegnimento del sistema	
2	Server	Verifica la presenza di connessioni aperte o di sessioni utente attive. Termina l'esecuzione del sistema.	
Scenario/Flusso di eventi Alternativo: Sono presenti connessioni aperte e l'admin procede con l'arresto forzato del sistema.			
2.a1	Server	Non effettua lo spegnimento del sistema. Avvisa l'amministratore dell'esistenza di connessioni ancora aperte mostrandogli un messaggio dove può scegliere di attendere che gli utenti concludano le loro attività o procedere con l'arresto forzato.	
2.a2	Admin	L'amministratore seleziona di procedere con l'arresto forzato del sistema.	
2.a3	Server	Interrompe bruscamente tutte le connessioni aperte notificando agli utenti l'interruzione del servizio.	
Scenario/Flusso di eventi di Errore: Sono presenti connessioni aperte e l'admin attende che gli utenti concludono la loro sessione.			
2.b1	Server	Non effettua lo spegnimento del sistema e notifica l'amministratore con un messaggio.	
2.b2	Admin	Decide di attendere che tutte le connessioni siano chiuse	
2.b3	Server	Il server interrompe gradualmente il sistema, assicurandosi di completare eventuali processi in corso e chiudendo connessioni in modo sicuro.	



		Una volta che tutte le connessioni sono chiuse, il sistema viene spento
--	--	---

Identificativo UC_BC_3	<i>Fallimento del sistema causa sovraccarico del server</i>	<i>Data</i>	04/12/2023
		<i>Vers.</i>	1.0
		<i>Autore</i>	Simone Vittoria
Descrizione	Il seguente BC descrive il comportamento del sistema a seguito di un fallimento durante la sua esecuzione.		
Attore Principale	Admin		
Attori secondari	Qualsiasi attore ipotizzato in tutta la progettazione (Admin, Magazziniere, Gestore Finanze, Cassiere), quindi per generalizzare si usa Dipendente.		
Entry Condition	Il server è in funzione AND Il sistema è in esecuzione sul server AND Un aumento significativo di utilizzo delle risorse e dei servizi del sistema porta ad un sovraccarico del server		
Exit condition On success	Il sistema riprende a funzionare correttamente		
Exit condition On failure	L'errore individuato non è l'errore che ha causato il fallimento del sistema AND Il sistema viene riavviato ma continua a non funzionare correttamente		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Dipendente	Riscontra un rallentamento e un calo precipitoso delle prestazioni del sistema. Alcuni servizi smettono di funzionare correttamente, non eseguendo le operazioni assegnate. Segnala all'amministratore del sistema quanto sta accadendo.
2	Admin	Utilizza sistemi di monitoraggio e analizza i file di log per indentificare la causa dell'errore. Individuato l'errore, informa gli utenti del disservizio temporaneo. Allo scopo di ristabilire la situazione, adotta misure di bilanciamento del carico e, successivamente, procede con il riavvio del server.



4

Servizi dei sottosistemi

In questa sezione, verranno elencati e descritti i servizi di ogni sottosistema individuato precedentemente.

Autenticazione

Servizio	Descrizione	Interfaccia
Login	Gestisce la funzionalità che permette ai dipendenti di accedere al sistema	Autenticazione Service
Logout	Gestisce la funzionalità che permette ai dipendenti di uscire dal sistema	Autenticazione Service
Modifica pin di accesso	Gestisce la funzionalità che permette all'admin di modificare i pin di accesso dei dipendenti	Autenticazione Service

Gestione Finanze

Servizio	Descrizione	Interfaccia
Visualizza bilancio	Gestisce la funzionalità per visualizzare i bilanci dell'attività	Gestione Finanze Service
Visualizza andamento prodotto	Gestisce la funzionalità di visualizzare l'andamento di un prodotto in termini di vendite	Gestione Finanze Service
Imposta sconti	Gestisce la funzionalità che permette di impostare uno sconto ad un prodotto per un periodo limitato	Gestione Finanze Service
Visualizza prodotti	Gestisce la funzionalità che permette di visualizzare l'elenco di tutti i prodotti	Gestione Finanze Service



Gestione Magazzino

Servizio	Descrizione	Interfaccia
Modifica fornitura	Gestisce la funzionalità che permette di modificare una fornitura all'admin precedentemente registrata	Gestione Magazzino Service
Registrazione fornitura	Gestisce la funzionalità che permette la registrazione di una nuova fornitura	Gestione Magazzino Service
Visualizza prodotti magazzino	Gestisce la funzionalità che permette di visualizzare i prodotti presenti in magazzino	Gestione Magazzino Service
Visualizza forniture	Gestisce la funzionalità che permette di visualizzare tutte le forniture registrate	Gestione Magazzino Service
Eliminazione lotti	Gestisce la funzionalità che permette la rimozione dei lotti di un determinato prodotto	Gestione Magazzino Service

Gestione Scaffali

Servizio	Descrizione	Interfaccia
Aggiungi prodotto scaffali	Gestisce la funzionalità che permette di aggiungere una certa quantità di un prodotto agli scaffali	Gestione Scaffali Service
Elimina scaduti scaffali	Gestisce la funzionalità che permette di rimuovere i prodotti scaduti dagli scaffali	Gestione Scaffali Service
Visualizza prodotti scaffali	Gestisce la funzionalità che permette di visualizzare i prodotti presenti sugli scaffali divisi prima per prodotti e poi per scadenze	Gestione Scaffali Service

Gestione Vendite

Servizio	Descrizione	Interfaccia
Vendita prodotti	Gestisce la funzionalità che permette di vendere i prodotti alla clientela	Gestione Vendite Service



Visualizza storico vendite

Gestisce la funzionalità che permette
di visualizzare lo storico delle vendite
effettuate

Gestione Vendite Service



Sigla/Termine	Definizione
DBMS	Database Management System è uno strato software che si frappone fra l'utente ed i dati veri e propri. Grazie a questo strato intermedio l'utente e le applicazioni non accedono ai dati così come sono memorizzati effettivamente, cioè alla loro rappresentazione fisica, ma ne vedono solamente una rappresentazione logica. Ciò permette un elevato grado di indipendenza fra le applicazioni e la memorizzazione fisica dei dati.
Modello ER	Modello Entity – Relationship dedicato alla rappresentazione della realtà in fase di sviluppo e allo sviluppo del database.
Piattaforma	Base software o hardware sulla quale è sviluppata o in esecuzione l'applicazione.
Lotto	Insieme di stessi prodotti con uguale data di scadenza arrivati da una determinata fornitura.
Fornitura	Rappresentazione in forma digitale dell'arrivo di lotti di vari prodotti con relative quantità scaricate.
Vendita	Riferimento digitale di vendite di prodotti avvenute nel market.